



2009 年度下期未踏 IT 人材発掘・育成事業 採択案件評価書

1. 担当PM

筧 捷彦 PM(早稲田大学 理工学術院基幹理工学部 情報理工学科 教授)

2. 採択者氏名

チーフクリエイター: 千々和 大輝(自由ヶ丘高等学校)

コクリエイター: なし

3. プロジェクト管理組織

株式会社オープンテクノロジーズ

4. 委託金支払額

1,962,825 円

5. テーマ名

従来のシステムと比べて柔軟性の高い key-value store の開発

6. 関連Webサイト

<http://github.com/daiki41ti/kyeeva>

7. テーマ概要

近年、クラウドコンピューティングにおける基盤技術として、key-value store が注目されている。このシステムは RDB と比べて大規模なクラスタ内での使用を前提としており、ノード数の増加に対応して自動的にスケールアウトするものがほとんどである。さらに、高速に動作する必要があるので RDB ほど処理が多様でない点も特徴的である。

提案者も今年の六月に、P2P 構造化オーバーレイネットワークの一つである Skip Graph を利用した簡易的な分散 key-value store を並列プログラミング言語 Erlang で実装している。このシステムの特徴は、大規模クラスタ内で効率的な範囲検索を行うことができるという点である。これは Skip Graph が Chord などの DHT とは異なり、ピアが key によってソートされているという性質を利用して実現している。

しかし一般的な key-value store は単一 key の get と put、delete のような既実装されている機能しか行うことができず、とても柔軟性があるとは言い難い。そこで本プロジェクトでは、以前作ったシステムを C++ で書き直して十分に高速化し、柔軟性向上のため以下の 2 つの新しい機能を備えた分散 key-value store を開発する。

1. value による範囲検索
2. 自律的な分散エージェントによる拡張機構

さらに、Skip Graph は DHT に劣らない検索効率でありながら範囲検索も行うことができるというとても優秀なアルゴリズムであるが、素直に実装すると物理ノード間のデータ分布が偏ってしまうという欠点が存在する。しかし、これはデータの put 時に DHT を利用してノードを選択することにより解消できると考えた。よって本プロジェクトでは並行してデータ分散基盤としての DHT の実装も行う。

成果物はオープンソースソフトウェアとしての公開を予定している。ライセンスは特に考えていないのでとりあえず BSD License とするつもりであるが、今後さらに緩いライセンスに変更する可能性もある。

8. 採択理由

クリエイターは、高校1年生(15歳)。提案は、Web キャッシュの手法として注目されている、key-value store の機構を、分散ハッシュテーブルの上で skip graph によって実現しようというもの。具体的には、隣接のピアの情報を保持しておき、分散エージェントとして別プロトコルで通信を行うことで拡張性をもたせるというものである。すでに、Erlang を用いてネットワーク上で実験的なシステムを動かしてみた経験をもつ。現在、それを C++ で書き直しているところだという。

この年齢でここまでのアイデアをもち、さらに手が動かせるというのはすばらしい。プログラミングは中学2年のときに始めたというが、驚くべき勉強ぶりである。提案内容の実現にとどまらず、この未踏ユースの期間にその能力を一気に開花させて、より高いレベルの成果を残してくれることを期待している。

9. 開発目標

このプロジェクト開始前にクリエイターは、P2P 構造化オーバーレイネットワークの一つである Skip Graph を利用した簡易的な分散 KVS を、並列指向関数型プログラミング言語 Erlang で実装していた。このシステムは、大規模クラスタ内で効率的な範囲検索を行うことができるという特徴がある。Skip Graph は、Chord などの DHT (Distributed Hash Table、分散ハッシュテーブル)とは異なり、ピアが key によってソートされているという性質を利用してのものである。

一般的な分散 KVS は単一キーを対象とした処理しか行うことができず、柔軟性に欠ける。このプロジェクトでは開発済みの分散 KVS を発展させ、柔軟性向上のためのつぎの二つの新しい機能を備えた分散 KVS を開発することを目標とした。

1. 範囲検索機能
2. 多次元構造

ところで、Skip Graph には、素直に実装すると物理ノード間のデータ分布が偏ってしまうという欠点が存在する。そこで、データの Put 時に DHT を利用してノードを選択することによってこの欠点を取り除くことにした。

10. 進捗概要

高校生であることから、プロジェクトに集中して割ける時間が思うようにはとれず、時間配分の難しいプロジェクトとなった。このプロジェクトの遂行に合わせて、まず、すでに作ってあった Skip Graph による分散 KVS を整理し直すこと、DHT を Erlang で実装することに着手した。その上で、範囲検索機能を持ち、多次元構造を提供するための仕組み作りに入った。途中で、ブースト会議による他クリエイターとの交流や、この分野の研究者との意見交換ができる機会をいくつか得て、基本着想の整理や実装を進めていった。

実装は、比較的順調に終わることができたが、その性能評価を十分に行うことや、成果報告会に向けての準備、最終報告書のまとめ等の他者への説明の準備を十分に行うことができなかつたのが心残りである。

11. 成果

既存の P2P (Peer to Peer) 技術を応用して従来の非集中型分散 KVS に内在する

「効率の良い範囲検索が難しい」「問い合わせ形式が単一のキーと値のペアに制

限される」という問題を解決することができた。

類似システムとしては、Google 社の BigTable、Facebook 社の Cassandra 等の分散 KVS が挙げられる。BigTable は、範囲検索が可能であるものの、管理ノードが必要という点で本システムと異なる。Cassandra は、管理ノードが不要であるものの、範囲検索に向いていないという点で本システムとは異なる。

開発したシステムについては、StarBED 上の 40 台のノードでの実証も行い、この分野の研究者が集う会議で発表を行う機会も得た。

12. プロジェクト評価

高校2年生(開発終了時)で、これだけの先端的テーマに取り組み、きちんと仕上げることができたのはすばらしい。Erlang という新しいプログラミング言語と C 言語とを組み合わせて、機能面だけでなく、性能向上を試みるなど、高いプログラミング能力を示すことができた。

当初の計画では、機能として範囲検索と分散配置を対象とするシステムを想定し、それを各種のプログラミング言語から使える形にライブラリ化することとしていた。しかしながら、この分野の研究者との意見交換から、多次元データに対する範囲検索を可能とする機能を提供する課題に取り組むのが、より重要な課題であることがわかり、これを開発目標に加え、ライブラリ化は見送ることとなった。

結果として、プロジェクト終了時に開発ができたものは、機能的に整えられたものとなったが、計画時に望んだとおりの実際に広く使ってもらうレベルには達しなかった。

13. 今後の課題

性能評価を大量データに対して行い、性能を上げる作業がある。特に、新ノードの Join 処理が遅いため、その高速化を図ること、また、同ノード上に存在するデータに対して範囲検索をかけることで、通信量を大幅に減らす可能性を追求するという課題がある。

広く使ってもらうための努力も課題である。まず、様々なプログラミング言語から使える形にライブラリ化をはかること、ドキュメントを整備することが重要である。さらには、システムの実用化に向けては、企業との連携もはかるなどのこともできるとよい。